M32C/80 Series

Using DMACII (Chained Transfer)

1. Abstract

This application note describes how to use DMACII in chained transfer.

2. Introduction

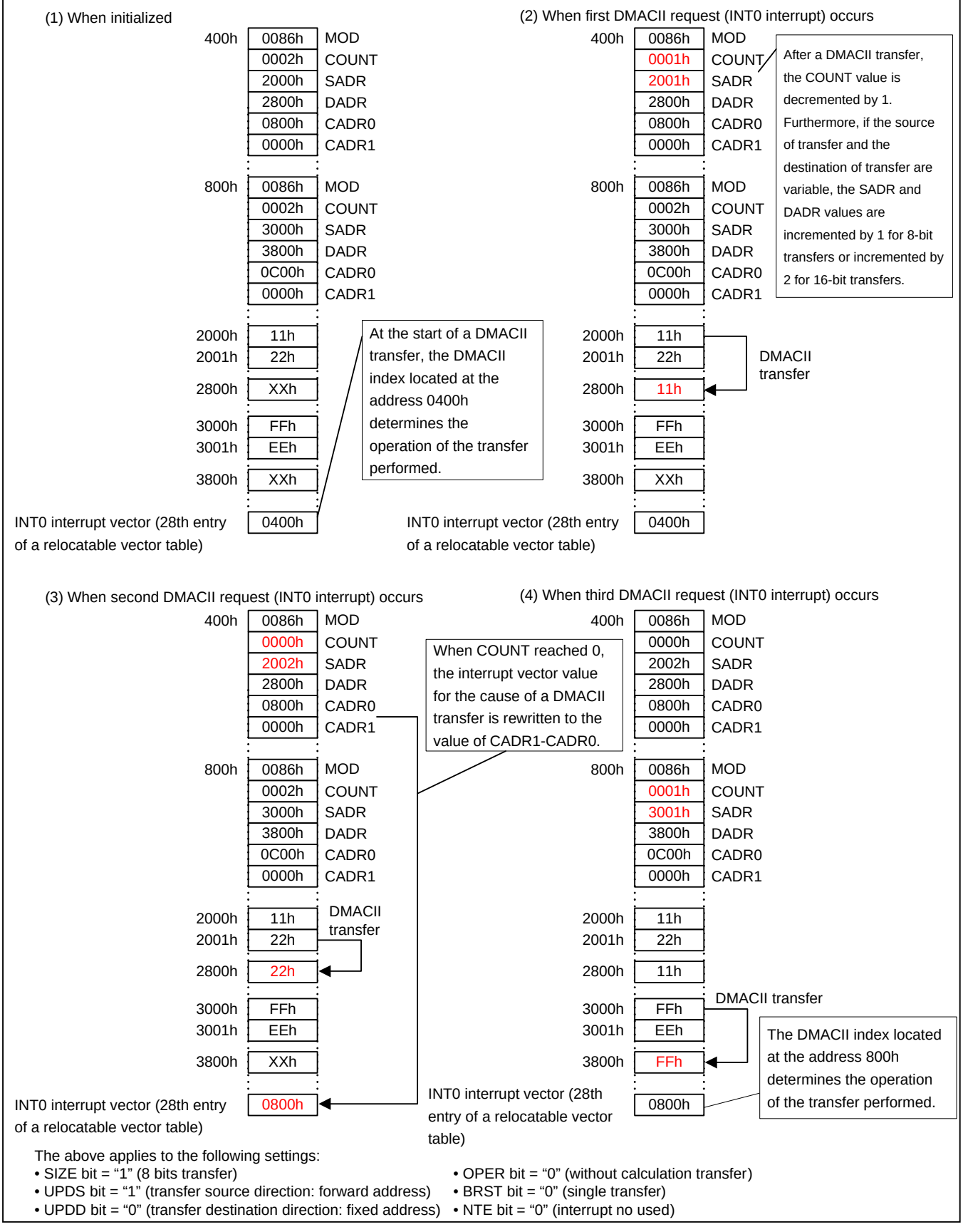
The explanation of this issue is applied to the following condition:
Applicable MCU: M32C/80 Series

This program can also be used when operating other microcomputers within the M16C family, provided they have DMACII function. However, some functions may have been modified.
Refer to the User's Manual for details. Use functions covered in this Application Note only after careful evaluation.

3. Detailed description

The following explains an example use of DMACII transfer for the case where each time an interrupt request which has had its priority level set to 7 occurs, data is transferred from memory to memory by a DMACII transfer and when a specified number of transfers has finished, the transfer mode, transfer count, source of transfer address, and destination of transfer address are changed.
Figure 1 shows a typical operation of a chained transfer mode.

Typical operation of a chain transfer in cases where the INT0 interrupt is used as the cause of DMACII



The above applies to the following settings:
- SIZE bit = “1” (8 bits transfer)
- UPDS bit = “1” (transfer source direction: forward address)
- UPDD bit = “0” (transfer destination direction: fixed address)
- OPER bit = “0” (without calculation transfer)
- BRST bit = “0” (single transfer)
- NTE bit = “0” (interrupt no used)

Figure 1. Typical Operation of a DMACII Chained Transfer Mode

3.1 DMAC II Transfer Mode

This application note example offers functions of chained transfer mode shown in Table 1.

Table 1. Selectable Functions in Chained Transfer Mode

Item	Definition	Selection
Transfer Block	8 bits	Yes
	16 bits	
Transfer Data	Immediate data	
	Data in memory	Yes
Source Direction	Fixed address	
	Forward address	Yes
Destination Direction	Fixed address	Yes
	Forward address	
Calculation Transfer Function	Without Calculation Transfer Function	Yes
	With Calculation Transfer Function	
Burst Transfer	Single transfer	Yes
	Burst transfer	
End-of-Transfer Interrupt	Interrupts not used	Yes
	Interrupts used	

3.2 DMAC II Index

When chain transfers are used, the DMACII index has 4 bytes of CADR1–CADR0 (chain transfer addresses) added. The total number of bytes that comprise the DMACII index is 12 unless end-of-transfer interrupt is used.
Be sure to create as many DMACII indices as the number of times a chain transfer is to be performed.
The DMAC II index must be located on the RAM area.

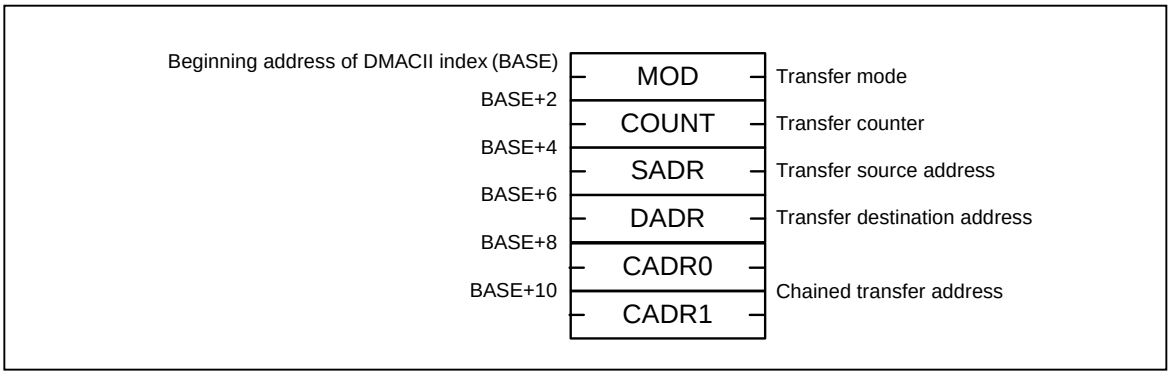


Figure 2.DMAC II Index

3.3 DMAC II Transfer

The interrupt requests from all peripheral functions whose ILVL2–ILVL0 bits in the interrupt control register have been set to “111b” constitute the cause of requests to DMAC II. In this application note, the INT0 interrupt is used for the cause of DMAC II request.

3.4 Setting Up the Relocatable Vector Table

The relocatable vector tables must be located on the RAM area.
During a chain transfer, when COUNT (transfer counter) reaches “0”, the peripheral function interrupt vector that constitutes the cause of a DMACII request changes to the CADR1–CADR0 value of the DMACII index.

3.5 Register Setting

To enable the operation defined in "Section 3. Detailed description", the following register settings must be taken place step by step. For detail configuration of each register, please refer to M32C/80 Series HARDWARE MANUAL.

(1) Interrupt disable

Set I flag to “0”.

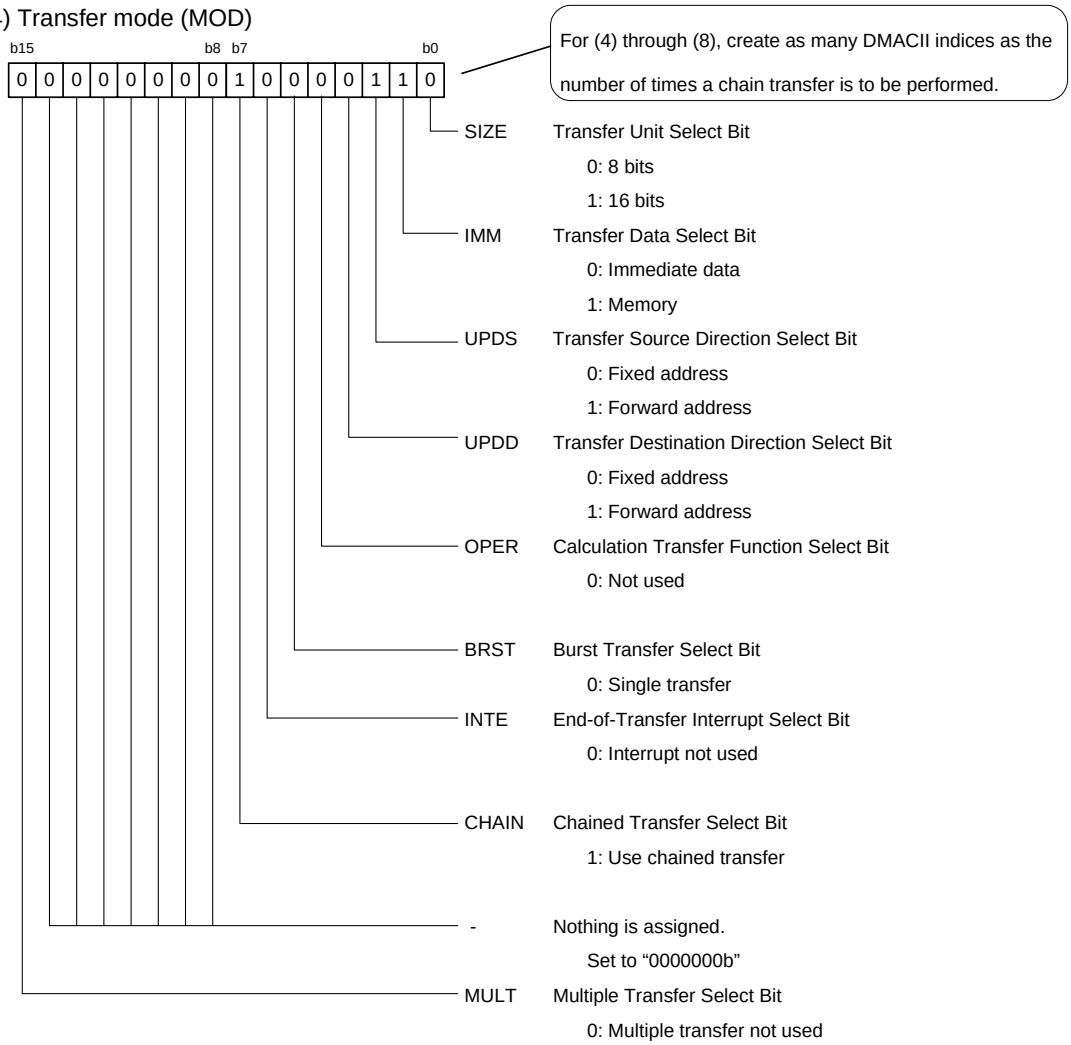
(2) Create a relocatable vector table

Create a relocatable vector table in the RAM.

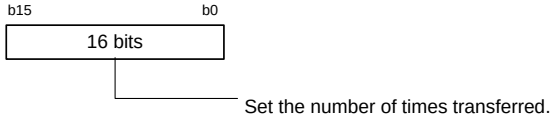
(3) Interrupt table register

Set the beginning address of the relocatable vector table located in the RAM.

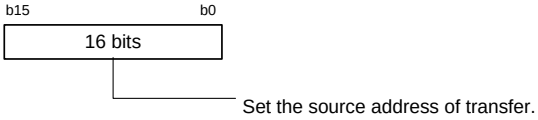
(4) Transfer mode (MOD)


b15
b8
b7
b0
0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0
For (4) through (8), create as many DMACII indices as the number of times a chain transfer is to be performed.
SIZE Transfer Unit Select Bit
0: 8 bits
1: 16 bits
IMM Transfer Data Select Bit
0: Immediate data
1: Memory
UPDS Transfer Source Direction Select Bit
0: Fixed address
1: Forward address
UPDD Transfer Destination Direction Select Bit
0: Fixed address
1: Forward address
OPER Calculation Transfer Function Select Bit
0: Not used
BRST Burst Transfer Select Bit
0: Single transfer
INTE End-of-Transfer Interrupt Select Bit
0: Interrupt not used
CHAIN Chained Transfer Select Bit
1: Use chained transfer
- Nothing is assigned.
Set to “0000000b”
MULT Multiple Transfer Select Bit
0: Multiple transfer not used

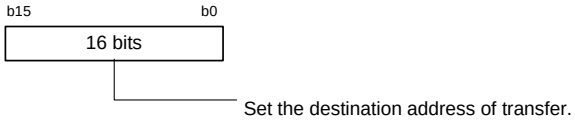
(5) Transfer count (COUNT)



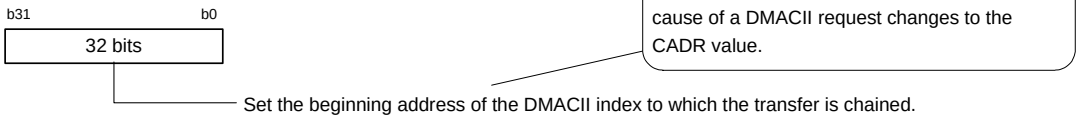
(6) Transfer source address (SADR)



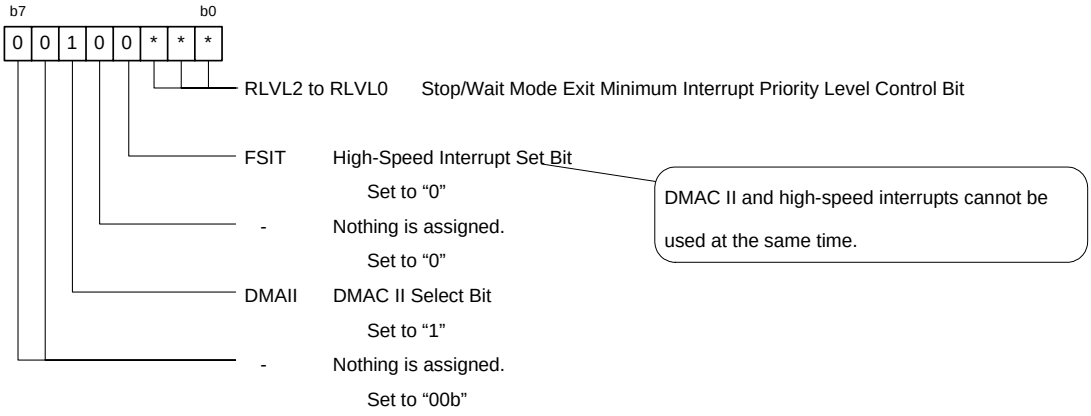
(7) Transfer destination address (DADR)



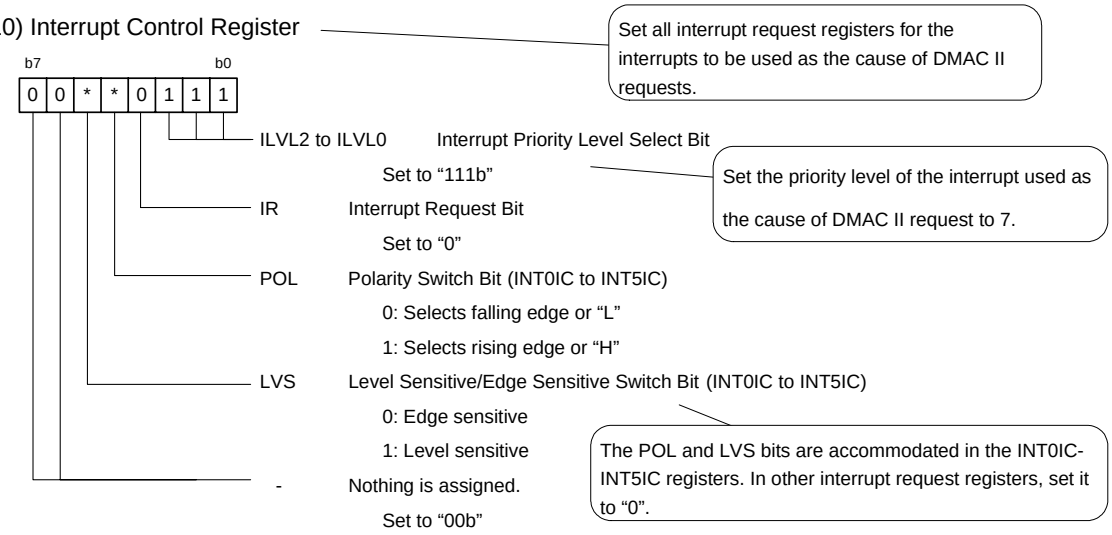
(8) Chained transfer address (CADR)


b31
b0
32 bits
When COUNT reaches “0”, the peripheral function interrupt vector that constitutes the cause of a DMACII request changes to the CADR value.
Set the beginning address of the DMACII index to which the transfer is chained.

(9) RLVL register


b7
b0
0 0 1 0 0 * * *
RLVL2 to RLVL0 Stop/Wait Mode Exit Minimum Interrupt Priority Level Control Bit
FSIT High-Speed Interrupt Set Bit
Set to “0”
- Nothing is assigned.
Set to “0”
DMAII DMAC II Select Bit
Set to “1”
- Nothing is assigned.
Set to “00b”
DMAC II and high-speed interrupts cannot be used at the same time.

(10) Interrupt Control Register


b7
b0
0 0 * * 0 1 1 1
Set all interrupt request registers for the interrupts to be used as the cause of DMAC II requests.
ILVL2 to ILVL0 Interrupt Priority Level Select Bit
Set to “111b”
IR Interrupt Request Bit
Set to “0”
POL Polarity Switch Bit (INT0IC to INT5IC)
0: Selects falling edge or “L”
1: Selects rising edge or “H”
LVS Level Sensitive/Edge Sensitive Switch Bit (INT0IC to INT5IC)
0: Edge sensitive
1: Level sensitive
- Nothing is assigned.
Set to “00b”
Set the priority level of the interrupt used as the cause of DMAC II request to 7.
The POL and LVS bits are accommodated in the INT0IC-INT5IC registers. In other interrupt request registers, set it to “0”.

4. Example of a Sample Program

4.1 C language source

```
/************************************************************************
/*  FILENAME: rej05b0640_src.c                                          *
/*  Ver : 1.00                                                        *
/*  FUNCTION: DMACII(Chained transfer)                                    *
/************************************************************************/
/******************************/
/*   include file            */
/******************************/
#include <stdio.h>
#include <string.h>
#include "sfr32c83.h"

/******************************/
/*   DMACII                   */
/******************************/
struct DMACII_INDEX{
   union   {
      struct{
         char    size:1;        /* Transfer Unit Select Bit */
         char    imm:1;         /* Transfer Data Select Bit */
         char    upds:1;        /* Transfer Source Direction Select Bit */
         char    updd:1;        /* Transfer Destination Direction Select Bit */
         char    oper:1;        /* Calculation Transfer Function Select Bit */
         char    brst:1;        /* Burst Transfer Select Bit */
         char    inte:1;        /* End-of-Transfer Interrupt Select Bit */
         char    chain:1;       /* Chained Transfer Select Bit */
         char    reserve:7;
         char    mult:1;        /* Multiple Transfer Select Bit */
      }mod_bit;
      unsigned short mod_word;
   }mod;
   unsigned short count;          /* Transfer count */
   unsigned char near *sadr;      /* Transfer source address */
   char near *dadr;              /* Transfer destination address */
   struct DMACII_INDEX far *cadr;  /* Chained transfer address */
}dm_index[2];

long ram_vect[64];            /* Relocatable vector table */

/*  Transfer data array  */
static unsigned char near data1[5] = {0x11,0x22,0x33,0x44,0x55};
static unsigned char near data2[5] = {0xff,0xee,0xdd,0xcc,0xbb};
/*  Transfer destination memory    */
unsigned char near dest1;
unsigned char near dest2;

#define S_VECTOR    ((unsigned long*)0xfefd00)  /* Relocatable vector address */

/******************************/
/*   main                     */
/******************************/
void main(void){
   asm("   fclr    i   ");    /* Interrupt disable */

   memcpy(ram_vect,S_VECTOR,64*4);    /* Copy relocatable vector to RAM */
   asm("   ldc     #_ram_vect, intb    "); /* Set relocatable vector to RAM */
   /*  DMACII setting  */
   dm_index[0].mod.mod_word = 0x0086;  /* Transfer Unit:      8bit    */
                               /* Transfer Data:      Memory  */
                               /* Transfer Source:    Forward */
                               /* Transfer Destination:Fixed   */
                               /* Calculation Transfer:None    */
                               /* Burst Transfer     :Single  */
                               /* Interrupt:         None    */
                               /* Chained Transfer:    Have    */
                               /* Multiple Transfer:   None    */

   dm_index[0].count = 5;            /* number of transfer = 5 */
   dm_index[0].sadr = data1;          /* Source of transfer  = beginning address of the data array  */
   dm_index[0].dadr = &dest1;         /* Destination of transfer = dest1 */
   dm_index[0].cadr = &dm_index[1];
```

```
    dm_index[1].mod.mod_word = 0x0006;  /* Transfer Unit:      8bit    */
                                /* Transfer Data:      Memory  */
                                /* Transfer Source:    Forward */
                                /* Transfer Destination:Fixed   */
                                /* Calculation Transfer:None    */
                                /* Burst Transfer     :Single  */
                                /* Interrupt:         None    */
                                /* Chained Transfer:   None    */
                                /* Multiple Transfer:  None    */

    dm_index[1].count = 5;         /* number of transfer = 5 */
    dm_index[1].sadr = data2;      /* Source of transfer  = beginning address of the data array */
    dm_index[1].dadr = &dest2;     /* Destination of transfer = dest2 */
    dm_index[1].cadr = &dm_index[0];

    /*  Set the interrupt used for DMAC II  */
    rlvl = 0x20;              /* Interrupt priority level 7 is used for DMAC II transfers */

    int0ic = 0x07;            /* INT0 interrupt level 7 (used for DMACII) */

    while(1);
}
```

4.2 Relocatable Vector Tables

```
    ;-----------------------------------------------------------------
    ; variable vector section
    ;-----------------------------------------------------------------
             .section  vector,ROMDATA              ; variable vector table
             .org      VECTOR_ADR

             .lword    dummy_int          ; BRK (software int 0)
             .lword    dummy_int          ;
             .lword    dummy_int          ;
             .lword    dummy_int          ;
             .lword    dummy_int          ;
             .lword    dummy_int          ;
             .lword    dummy_int          ;
             .lword    dummy_int          ;
             .lword    dummy_int          ; DMA0 (software int 8)
    ;
    ;
    ;                          (Omission)
    ;
    ;
             .lword    dummy_int          ; INT5 (software int 26)
             .lword    dummy_int          ; INT4 (software int 27)
             .lword    dummy_int          ; INT3 (software int 28)
             .lword    dummy_int          ; INT2 (software int 29)
             .lword    dummy_int          ; INT1 (software int 30)
             .glb      _dm_index
             .lword    _dm_index          ; INT0 (software int 31)
             .lword    dummy_int          ; TIMER B5 (software int 32)
    ;
    ;
    ;                          (Omission)
    ;
    ;
```

5. Reference

Renesas Technology Corporation Home Page

http://www.renesas.com/

E-mail Support

E-mail: csc@renesas.com

Hardware Manual

M32C/80 Group Hardware Manual
(Use the latest version on the home page: http://www.renesas.com)

TECHNICAL UPDATE/TECHNICAL NEWS

(Use the latest information on the home page: http://www.renesas.com)

REVISION HISTORY

Rev.	Date	Description	
		Page	Summary
1.00	2005.05.20	-	First edition issued

Keep safety first in your circuit designs!

1. Renesas Technology Corporation puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corporation product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corporation or a third party.
2. Renesas Technology Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors.
   Renesas Technology Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corporation by various means, including the Renesas Technology Corporation Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corporation for further details on these materials or the products contained therein.